2016 IEEE 6th International Conference on Advanced Computing

Multi-Number CVT-XOR Arithmetic Operations In Any Base System And Its Significant Properties

Jayanta Kumar Das
Applied Statistics Unit
Indian Statistical Institute
Kolkata-700108, India
dasjayantakumar89@gmail.com

Pabitra Pal Choudhury
Applied Statistics Unit
Indian Statistical Institute
Kolkata-700108, India
pabitrapalchoudhury@gmail.com

Sudhakar Sahoo
Faculty of Computer Science
Institute of Mathematics and Applications
Bhubaneswar-751003, India
sudhakar.sahoo@gmail.com

***Abstract*—Carry Value Transformation (CVT) is one of the modified structures of Integral Value Transformations (IVTs) and coming under the category of discrete dynamical system. Earlier in [5] it has been proved that the addition of two non-negative integers is equal to the addition of their CVT values and XOR values and is true in any base of the number system. In the present study, this phenomenon is extended to perform CVT and XOR operations for many non-negative integers in any base system. To achieve this both the definition of CVT and XOR are modified over the set of multiple integers instead of two. Also some important properties of these operations have been studied. With the help of cellular automata the adder circuit designed in [14] on using CVT-XOR recurrence formula is used to design a parallel adder circuit for multiple numbers in binary number system.**

***Keywords*-Integral Value Transformations; Carry Value Transformation; Recursion; Adder circuit;**

I. INTRODUCTION

Integral Value Transformations (IVTs) is a class of continuous maps in a discrete space and was introduced in the year 2009 $[1,2,7]$. A p-adic, k-dimensional, Integral Value Transformation is denoted by $IVT_j^{p,k}$ and it is a mapping from $N_0^k \longrightarrow N_0$. When k=1, $IVT_j^{p,1}$ is defined as

$IVT_j^{p,1}(X) = (f_j(x_n)f_j(x_{n-1})...f_j(x_1))_p = (m)_{10}$
where X is a non-negative p-adic integer represented as $X = (x_nx_{n-1}...x_1)_p$, m is a decimal number after conversion from its corresponding p-adic number and the rule number f_i is a local mapping defined from the set $\{0,1,2,3,...,p-1\}$ to itself. Here j is also a decimal number of the p-adic string in the truth table representation of the local map.

For example, when $p = 3$, $k = 1$ and say $X = (14)_{10} = (112)_3$ and for the two different rule numbers 5 and 16 shown in TABLE I, the $IVTs$ are calculated as $IVT_5^{3,1}(14) = (f_5(1)f_5(1)f_5(2))_3 = (110)_3 = (12)_{10}$ and $IVT_{16}^{3,1}(14) = (f_{16}(1)f_{16}(1)f_{16}(2))_3 = (221)_3 = (25)_{10}$

Like One dimensional, two dimensional p-adic, rule j IVT denoted by $IVT_j^{p,2}(X,Y)$ is defined as $IVT_j^{p,2}(X,Y) = (f_j(x_n,y_n)f_j(x_{n-1},y_{n-1})...f_j(x_1,y_1))_p = (m)_{10}$. Similarly, k-dimensional $IVTs$ can be defined. (Sometimes the symbol β is used instead of p as base of the number system)

TABLE I TRUTH TABLE OF TWO 1-VARIABLE TERNARY (BASE-3) FUNCTIONS f_5 AND f_{16}

Variable	Rule	
x_i	$\mathbf{f_5}$	$\mathbf{f_{16}}$
0	2	1
1	1	2
2	0	1

Carry Value Transformation (CVT) which was initially defined in the year 2008 $[3]$, later developed and elaborated in $[4,5]$ became a special case of $IVT_j^{p,k}(X,Y)$ when $p = 2$, $k = 2$ and $j = 8$ along with a 0 padded in the LSB position of the output binary string. Thus CVT is a two dimensional, rule 8, binary IVT with a 0 padded in it where as XOR is simply a two dimensional rule 6 binary IVT as shown in TABLE II.

TABLE II TRUTH TABLE OF TWO 2-VARIABLE TERNARY (BASE-3) FUNCTIONS f_8 and f_6

Variable	Rule	
x_i y_i	$\mathbf{f_8}$	$\mathbf{f_6}$
0 0	0	0
0 1	0	1
1 0	0	1
1 1	1	0

Carry Value Transformations were studied in $[3]$ for producing self-similar fractal whose dimension is exactly same as the dimension of the Sierpinski triangle and structure very similar to it. Further they have shown that CVT can also be used to produce many periodic and chaotic patterns. Also, analytical and algebraic properties of CVT were studied in $[3]$. Different fractals having dimension in between 1 and 2 were studied in $[4]$. Two most important properties of CVT and Modied Carry Value Transformation (MCVT) were studied in $[5]$. Where It has been shown that (1) if X and Y are two non-negative integers then their sum is equal to their CVT and XOR values i.e.

978-1-4673-8286-1/16 $31.00 © 2016 IEEE
DOI 10.1109/IACC.2016.147

$X + Y = CVT(X, Y) + XOR(X, Y)$ and not only binary this is a recursive scheme and is valid even if X and Y are expressed using any base (2) in case of binary, the number of iterations leading to either $CVT = 0$ or $XOR = 0$ does not exceed the maximum of the lengths of the two numbers X and Y both expressed in binary i.e. the convergence property of CVT, XOR and $MCVT$ were also discussed.

Other IVTs called Extreme Value Transformation (EVT) [4], $2-$ Variable Boolean Operation (2-VBO) [5] are also defined and are used to manipulate binary strings. Its application in the formation of patterns $[4, 7]$, solving Round Rabin Tournament problem [8], mathematical genomics [16] etc. are also studied. The Collatz-like IVTs which converges to zero (a fixed point) are rigorously discussed in $[7, 15]$. Previously designed adder circuits $[9, 10, 11, 12]$ are combinational in nature and their hardware complexity depends on number of logic gates used in the circuit and the associated gate delays. In line with this, Cellular Automata Machines (CAMs) [13] were studied in [14], for efficient hardware design of some basic arithmetic operations like addition, subtraction, multiplication and division. The complexity of these circuits does not depend on the gate delays rather depends on number of clock cycles required to complete the computation.

The paper is organized in the following way: In section II some of the preliminary concepts on CVT and XOR operation of two numbers in binary domain is highlighted (also, thoroughly elaborated in [3]). In section III we have discussed the CVT and XOR operations of many numbers in any base system and studied some of its important properties. In section IV a parallel architecture for multi number addition in binary number system has been proposed. In Section V conclusion for this article along with some future research planning have been added.

II. CARRY VALUE TRANSFORMATION

The carry or overflow bits are usually generated at the time of addition between two n-bit strings. In the usual addition process, carry value which is same as the bit wise ANDing operation, is always a single bit and if generated then it is added column wise with other bits and not necessarily save for further use. But the carry value defined in [3] are the usual carries generated bit wise and stored in their respective places as shown in TABLE III. The value 0 in the LSB position denotes there is no carry on the right.

TABLE III GENERATED CARRY VALUES AND XOR VALUES ARE SHOWN WHILE ADDING TWO INTEGERS X AND Y EXPRESSED IN BINARY

carry value	=	$x_n \wedge y_n$	$x_{n-1} \wedge y_{n-1}$	$x_{n-2} \wedge y_{n-2}$	...	$x_1 \wedge y_1$	0
X	=		x_n	x_{n-1}	...		x_1
Y	=		y_n	y_{n-1}	...		y_1
$X \oplus Y$	=		$x_n \oplus y_n$	$x_{n-1} \oplus y_{n-1}$	...		$x_1 \oplus y_1$

Thus the corresponding decimal value of the string of carry bits is always an even integer.

Precise form of CVT is a mapping defined as $(B_n \times B_n)$ to B_n where B_n is the set of strings of length n on $B = \{0, 1\}$. More specifically, if $X = (x_n, x_{n-1}, ..., x_1)$ and $Y = (y_n, y_{n-1}, ..., y_1)$ then $CVT(X, Y) = (c_n, c_{n-1}, ..., c_1, 0)$ where $c_n = x_n \wedge y_n$ is an $n + 1$ bit strings, belonging to set of non-negative integers and can be computed bit wise by logical AND operation followed by a 0 which denotes no carry is generated in the LSB at the time of addition procedure.

Illustration 1. *Let us take two non-negative integers* 11 *and* 13 *and we wish to calculate its CVT and XOR value. The binary representation of* 11 *is* $(1011)_2$ *and* 13 *is* $(1101)_2$. *The carry value is calculated using bitwise ANDing operation followed by zero-padding and XOR is calculated using usual XOR operation defined for Boolean circuits and shown in TABLE IV:*

TABLE IV CVT AND XOR VALUES ARE CALCULATED FOR TWO POSITIVE INTEGERS 11 AND 13

CVT	:	1	0	0	1	0
11	:		1	0	1	1
13	:		1	1	0	1
XOR	:		0	1	1	0

Thus in our example, $CVT(11, 13) = CVT(1011, 1101)_2 = (10010)_2 = (18)_{10}$ and $XOR(11, 13) = XOR(1011, 1101)_2 = (0110)_2 = (6)_{10}$. Figure 1 shows the circuit diagram of CAM used for performing addition of two 4-bit numbers [14]. This CAM is based on a recurrence relation

$$X + Y = CVT(X, Y) + XOR(X, Y) \tag{1}$$

which has been proved to be valid for any base system [5]. In case of 11 and 13, the successive CVT and XOR value are calculated and recursively added and shown as: $11 + 13 = 18 + 6 = 4 + 20 = 16 + 8 = 0 + 24$ and in this case the recurrence relation requires 3 iterations to compute the sum. It can be noted that instead of 11 and 13 if one selects 18 and 6 the iteration required is 2 and for pairs 4 and 20 it takes one iteration to get the final sum 24. Thus the number of iteration to obtain the sum completely depends on the binary pattern of two integers supplied to the adder circuit. Therefore in a large scale computation the average circuit complexity varies on the distribution of integer pairs given to the circuit.

This CAM is used to design an adder circuit for multiple numbers in binary number system and proposed in section IV.

III. CVT AND XOR OPERATIONS OF MANY NUMBERS IN ANY BASE SYSTEM

Definition: For any number system (say in base β), XOR and CVT of K non-negative integers is defined as follows:

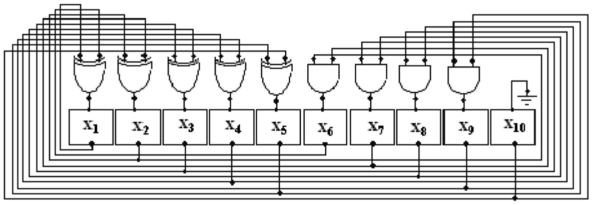

Figure 1 Figure shows the 4-bit adder circuit called CAM

(here K integers are symbolized as $X_1, X_2, X_3, ..., X_K$ and each are represented using n number of bits)

$X_1 = a_{1n}, ..., a_{11}$
$X_2 = a_{2n}, ..., a_{21}$
.
.
.
$X_K = a_{Kn}, ..., a_{K1}$

$XOR(X_1, X_2, X_3, ..., X_K) = (R_n, R_{n-1}, ..., R_1)_\beta$ and
$CVT(X_1, X_2, X_3, ..., X_K) = (C_n, C_{n-1}, ..., C_1, 0)_\beta$
where $C_i = \lfloor (a_{1i} + a_{2i} +, ..., +a_{Ki})/\beta \rfloor$
$R_i = (a_{1i} + a_{2i} + ... + a_{Ki}) \mod \beta$ and
$i = 1, 2, 3, ..., n.$

Note: In general, C_i result may not be in base β system but the decimal conversion of $(C_n, C_{n-1}, ..., C_1, 0) = (C_n \times \beta^{n-1} + C_{n-1} \times \beta^{n-2} + ... + C_1 \times \beta^1 + 0)$ is same as the method from base β to base 10.

Theorem 1. *The recurrence relation in equation (1) is also valid for many numbers in any base system.*
That is $X_1 + X_2 + X_3 + ... + X_K = CVT(X_1, X_2, X_3, ..., X_K) + XOR(X_1, X_2, X_3, ..., X_K)$

The proof of the above theorem can be similarly seen by extending the proof in [5].

Illustration 2. *Suppose in ternary number system (i.e.* $\beta = 3$*) CVT and XOR operation of decimal numbers* $17, 8, 11, 8, 4, 8$ *are as follows:*

$CVT(17, 8, 11, 8, 4, 8) = CVT(122, 022, 102, 022, 011, 022)$
$= (\lfloor 2/3 \rfloor \lfloor 9/3 \rfloor \lfloor 11/3 \rfloor 0) = (0330) = (0 \times 3^3 + 3 \times 3^2 + 3 \times 3^1 + 0 \times 3^0) = (36)_{10}$ *and*

$XOR(17, 8, 11, 8, 4, 8) = XOR(122, 022, 102, 022, 011, 022) = (202)_3 = (20)_{10}$.
So we have observed that $17 + 8 + 11 + 8 + 4 + 8 = CVT(17, 8, 11, 8, 4, 8) + XOR(17, 8, 11, 8, 4, 8) = 36 + 20.$

A. Important Corollaries in any Base System

Following two important Corollaries i.e. Corollary 1 and Corollary 2 are trivially obtained from the definitions of CVT and XOR operations for any base system and for arbitrary K numbers.

Corollary 1. From XOR definition, $R_i = \{0, 1, 2, ..., \beta - 1\}$ and $R_i = 0$ iff R_i is a multiple of β. So $XOR(X_1, X_2, ..., X_K)$=0 iff R_i is multiples of β $\forall i$.

Corollary 2. Similarly from CVT definition, $C_i = \{0, 1, 2 ... \frac{K \times (\beta - 1)}{\beta}\}$ and $C_i = 0$ iff $C_i < \beta$. So $CVT(X_1, X_2, ..., X_K)$=0 iff $C_i < \beta$ $\forall i$.

B. Important Properties of Multi Numbers CVT and XOR Operations in Binary Number System

1) CVT-Property:

Property 1. *For base* β *system, if all the* K *numbers are same then*

(a) *if* K *is even,* CVT *is equal to addition of* K *numbers i.e.* $CVT(X, X, ..., K$ *times*$) = X + X + ... + K$ *times*$= K \times X$.
(b) *if* K *is odd,* CVT *is equal to addition of* $(K - 1)$ *numbers i.e.* $CVT(X, X, ... K$ *times*$) = X + X + ... + (K - 1)$ *times*$= (K - 1) \times X$.

Illustration 3. *For base* $\beta = 2$ *and take X=5 (101)*
if K is even (say K=4) then
$CVT(5, 5, 5, 5) = CVT(101, 101, 101, 101) = (2020)_2 = (20)_{10} = 4 \times 5 = K \times X$
if K is odd (say K=5) then
$CVT(5, 5, 5, 5, 5) = CVT(101, 101, 101, 101, 101) = (2020)_2 = (20)_{10} = (5 - 1) \times 5 = (K - 1) \times X.$

Property 2. *For base* β *system, if* $CVT(X_1, X_2, ..., X_n) = P$ *and let* K *is a scalar quantity and is a power of* β *then*

(a) $CVT(K \times X_1, K \times X_2, ..., K \times X_n) = K \times P$ *and*
(b) $CVT(\frac{X_1}{K}, \frac{X_2}{K}, ..., \frac{X_n}{K}) = \lfloor \frac{P}{K+m} \rfloor$*; where* $m = \frac{\textit{number of odd numbers}}{\frac{K}{2}}$

Illustration 4. *For base* $\beta = 2$ *and let* $X_1 = 5(101), X_2 = 4(100), X_3 = 6(110), X_4 = 7(111)$ *and* $K = 4$.
$CVT(5, 4, 6, 7) = CVT(101, 100, 110, 111) = (2110)_2 = (22)_{10} = P$ *then*
$CVT(4 \times 5, 4 \times 4, 4 \times 6, 4 \times 7) = CVT(10100, 10000, 11000, 11100) = (211000)_2 = (88)_{10} = 4 \times 22 = K \times P$ *and*
$CVT(\frac{5}{4}, \frac{4}{4}, \frac{6}{4}, \frac{7}{4}) = CVT(1, 1, 1, 1) = CVT(1, 1, 1, 1) = (20)_2 = (4)_{10} = \lfloor \frac{22}{4+1} \rfloor = \lfloor \frac{P}{K+m} \rfloor$.

Property 3. *If* $CVT(X_1, X_2, ..., X_n) = P$ *and*
$CVT(Y_1, Y_2, ..., Y_n) = Q$ *then*
$CVT(X_1, X_2, ..., X_n, Y_1, Y_2, ..., Y_n) = P + Q$

Illustration 5. *For base* $\beta = 2$ *and let* $X_1 = 5(101), X_2 = 4(100), X_3 = 6(110), X_4 = 7(111)$ *and* $Y_1 = 13(1101), Y_2 = 9(1001), Y_3 = 9(1001), Y_4 = 13(1101)$.
$CVT(5, 4, 6, 7) = CVT(101, 100, 110, 111) = (2110)_2 =$

$(22)_{10} = P$ *and*
$CVT(13,9,9,13) = CVT(1101,1001,1001,1101) = (21020)_2 = (44)_{10} = Q$ *then*
$CVT(5,4,6,7,13,9,9,13) = CVT(0101,0100,0110,0111,1101,1001,1001,1101) = (23130)_2 = (66)_{10} = P+Q$.

Property 4. *If* $CVT(X,X,...,X) = P$ *then* $CVT(X^K,X^K,...,X^K) = P \times X^{k-1}$.

Illustration 6. *For base* $\beta = 2$ *and let* $X = 3(11)$, *K=3* $CVT(3,3,3,3) = CVT(11,11,11,11) = (220)_2 = (12)_{10} = P$ *then*
$CVT(3^3,3^3,3^3,3^3) = CVT(11011,11011,11011,11011) = (220220)_2 = (108)_{10} = 12 \times 3^{3-1} = P \times X^{K-1}$.

2) XOR-Property:

Property 5. *If all the* K *numbers are same then*

(a) *if* K *is even,* XOR *of* K *numbers is zero i.e.* $XOR(X,X,...K$ *times*$) = 0$.
(b) *if* K *is odd,* XOR *of* K *numbers is equal to a single number i.e.* $XOR(X,X,...K$ *times*$) = X$.

Illustration 7. *For base* $\beta = 2$ *and take X=5 (101)*
if K is even (say K=4) then
$XOR(5,5,5,5) = XOR(101,101,101,101) = (0)_2 = 0$
if K is odd (say K=5) then
$XOR(5,5,5,5,5) = XOR(101,101,101,101,101) = (101)_2 = (5)_{10} = X$.

Property 6. *For base* β *system, if* $XOR(X_1,X_2,...,X_n) = Q$ *and let* K *is a scalar quantity and is a power of* β *then*

(a) $XOR(K \times X_1, K \times X_2, ..., K \times X_n) = K \times Q$ *and*
(b) $XOR(\frac{X_1}{K},\frac{X_2}{K},...,\frac{X_n}{K}) = \lfloor \frac{Q}{K} \rfloor$.

Illustration 8. *For base* $\beta = 2$ *and let* $X_1 = 5(101), X_2 = 4(100), X_3 = 5(101), X_4 = 7(111)$ *and* $K = 4$.
$XOR(5,4,5,7) = XOR(101,100,101,111) = (011)_2 = (3)_{11} = Q$ *then*
$XOR(4 \times 5, 4 \times 4, 4 \times 5, 4 \times 7) = XOR(10100,10000,10100,11100) = (01100)_2 = (12)_{10} = 4 \times 3 = K \times Q$ *and*
$XOR(\frac{5}{4},\frac{4}{4},\frac{5}{4},\frac{7}{4}) = XOR(1,1,1,1) = XOR(1,1,1,1) = (0)_2 = (0)_{10} = \lfloor \frac{3}{4} \rfloor = \lfloor \frac{Q}{K} \rfloor$.

Property 7. *If* $XOR(X_1,X_2,...,X_n) = P$ *and* $XOR(Y_1,Y_2,...,Y_n) = Q$ *then*
$XOR(X_1,X_2,...,X_n,Y_1,Y_n,...,Y_n) = P \oplus Q$

Illustration 9. *For base* $\beta = 2$ *and let* $X_1 = 5(101), X_2 = 4(100), X_3 = 5(101), X_4 = 7(111)$ *and* $Y_1 = 13(1101), Y_2 = 9(1001), Y_3 = 9(1001), Y_4 = 10(1010)$.
$XOR(5,4,5,7) = XOR(101,100,101,111) = (011)_2 = (3)_{10} = P$ *and*
$XOR(13,9,9,10) = XOR(1101,1001,1001,1010) = (0111)_2 = (7)_{10} = Q$ *then*
$XOR(5,4,5,7,13,9,9,10) = XOR(0101,0100,0101,0111,1101,1001,1001,1010) = (0100)_2 = (4)_{10} = 3 \oplus 7 = P \oplus Q$.

Property 8. *If* $XOR(X,X,...,X) = P$ *then* $XOR(X^K,X^K,...,X^K) = P \times X^{K-1}$.

Illustration 10. *For base* $\beta = 2$ *and let* $X = 3(11)$, *K=2* $XOR(3,3,3) = XOR(11,11,11) = (11)_2 = (3)_{10} = P$ *then*
$XOR(3^2,3^2,3^2) = XOR(1001,1001,1001) = (1001)_2 = (9)_{10} = 3 \times 3^{2-1} = P \times X^{K-1}$.

IV. PROPOSED ADDER CIRCUIT FOR MULTIPLE NUMBERS IN BINARY NUMBER SYSTEM

The Figure 2 shows the circuit design for $K(=16)$ 4-bit numbers using CAMs. Where $K-1$ CAMs are required. For each CAM internal circuit design is throughly elaborated in [14] and also shown in Figure 1. Only difference is that number of inputs for each CAM is increased by 1 from first level to onwards. Initially in first level, computation is performed on 8 CAMs for two numbers pairwise $(x_1,x_2),(x_3,x_4),...,(x_{15},x_{16})$ in parallel. Output from each 8 CAMs are forwarded to the 4 second level CAMs and so on.

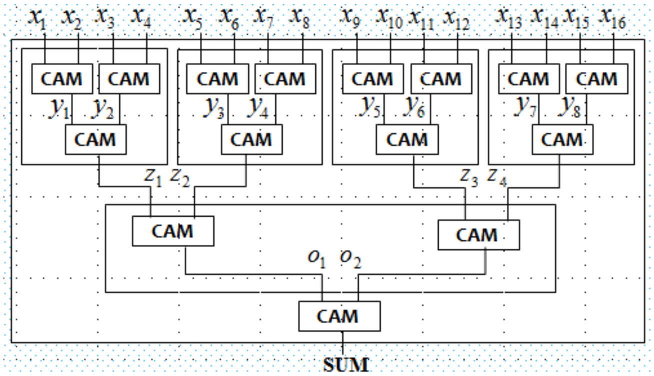

Figure 2 Circuit diagram for performing addition of sixteen 4-bit numbers in parallel using 15 CAMs

Worst case time complexity of parallel CAMs:
Let, n = all K integers are represented using n number of bits;
l = number of levels in circuit diagram, where $l = \lceil \log_2 K \rceil$ and T_C = Total worst case time complexity of parallel CAMs.
In the proposed CAMs, maximum delay in first level is $n+1$ unit (1 unit may be required for testing either CVT=0 or XOR=0), in second level is $n+2$ unit i.e. increased by 1 unit in each level and so on up-to last level.
Therefore, $T_C = \{(n+1)+(n+2)+\cdots+(n+l)\}$.

From Figure 2, (where $K = 16$, $n = 4$ and $l = 4$) maximum delay or worst case time complexity is $(4+1)+$

$(4+2)+(4+3)+(4+4)=26$ unit to perform addition of 16 integers.

V. CONCLUSION

Here we have seen how to perform the Multi-Number CVT and XOR Operation in any base system. Some important properties of these operations are highlighted both in any base and binary number system. The implementations of this multi number arithmetic operations in binary system using parallel adder circuit has been proposed. In this context another parallel adder circuit design can be performed on using recurrence relation where lesser number of CAMs are required compare to the circuit design shown in Figure 2. Further in future, we are trying to extend the CVT and XOR operations for fractional numbers and its significant properties.

ACKNOWLEDGMENT

The authors would like to thank Dr. Sk. Sarif Hassan and Mr. Suryakanta Pal for providing valuable suggestions to work in this domain.

REFERENCES

[1] Sk. S. Hassan, A. Ray, and P. Pal Choudhury, *Integral Value Transformation: A Class of Discrete Dynamical Systems*, Journal of Applied Mathematics and Computing, DOI: 10.1007/s12190-014-0825-y,09, 2014.

[2] Sk. S. Hassan, P. Pal Choudhury, B K Nayak, Avishek Ghosh, Joydeep Banerjee, *Integral Value Transformations: A Class of Affine Discrete Dynamical Systems and an Application* , Journal of Advanced Research in Applied Mathematics, vol-7, issue-1, pp-62-73, 2015.

[3] P. Pal Choudhury, S. Shaoo and B. K. Nayak, *Theory of Carry Value Transformation and its Application in Fractal formation*, IEEE International Advance Computing Conference,DOI: 10.1109/IADCC.2009.4809146, 2009.

[4] P. Pal Choudhury, Sk. S. Hassan, S. Shaoo and B. K. Nayak, *Act of CVT and EVT in the Formation of Number Theoretic Fractals*,International Journal of Computational Cognition (HTTP://WWW.IJCC.US), VOL. 9, NO. 1, 2011.

[5] S. Pal, S. Shaoo and B. K. Nayak, *Properties of Carry Value Transformation*, International Journal of Mathematics and Mathematical Sciences Volume 2012, Article ID 174372, 10 pages http://dx.doi.org/10.1155/2012/174372, 2012.

[6] S. Sahoo, I. Mohanty, G. Chowdhary and A. Panigrahi, *2-Variable Boolean Operation-Its Use in Pattern Formation*, arXiv: 1008.2530v1, nlin.CD.

[7] Sk. S. Hassan, P. Pal Choudhury, S. Das, R. Singh and B. K. Nayak, *Collatz function like Integral Value Transformations*, Alexandria journal of mathematics, Vol. 1, No. 2, Nov. 2010.

[8] P. Pal Choudhury, Sk. S. Hassan,Sahoo S. and B. K. Nayak, *Theory of Rule 6 and its Application to Round Robin Tournament*, arXiv 0906.5450v1, cs.DM, cs.GT, Int. Journal of Computational Cognition, Vol. 8, No. 3, pp. 33-37, Sep. 2010.

[9] J. L. Hennessy and D.A Patterson, *Computer Architecture: A Quantitative Approach* (2nd edition), Morgan Kaufmann, San Francisco, 1996.

[10] J. C. Lo, *A Fast Binary Adder with Conditional Carry Generation*, IEEE Transactions on Computers, Vol. 46, No. 2, pp. 248-253, Feb. 1997.

[11] T. Lynch, E. E. Swartzlander Jr. *A Spanning Tree Carry Lookahead Adder*, IEEE Transactions on Computers, Vol. 41, No. 8, pp. 931-939, 1992.

[12] J. M. Dobson and G. M. Blair, *Fast twos complement VLSI adder design*, Electronics Letters, Vol. 31, No. 20, pp.1721-1722, 28th September 1995.

[13] T. Toffoli, N. Margolis, *Cellular Automata Machines*, Cambridge, MA, MIT Press 1987.

[14] P. Pal Choudhury, S. Shaoo and M. Chakraborty, *Implementation of Basic Arithmetic Operations using Cellular Automata*, IEEE Computer Society,ISBN:978-0-7695-3513-5, http://doi.ieeecomputersociety.org/10.1109/ICIT.2008.18 pp: 79-80,Dec. 17, 2008 to Dec. 20, 2008.

[15] Sk. S. Hassan, *Discrete dynamics of one dimensional Collatz like integral value transformations*, Journal of Applied Mathematics and Computing, Vol. 49, Issue 1, pp 91-105, October 2015.

[16] Sk. S. Hassan, P. Pal Choudhury, B. S. Daya Sagar, S. Chakraborty, R. Guha, A. Goswami, *Quantitative description of genomic evolution of olfactory receptors*, Asian-European Journal of Mathematics, Volume 08, Issue 03, September 2015.